DRA/LANGLEY

Semi-Annual Report

Grant No. NAG-1-242 ✓

RESEARCH IN COMPUTER SCIENCE

Submitted to:

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Attention: Dr. John N. Shoosmith
ACD, MS 125

Submitted by:

James M. Ortega
Professor and Chairman

Report No. UVA/528209/AMCS85/105

July 1984

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF APPLIED MATHEMATICS

UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA 22901

Semi-Annual Report

Grant No. NAG-1-242

RESEARCH IN COMPUTER SCIENCE

Submitted to:

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Attention: Dr. John N. Shoosmith
ACD, MS 125

Submitted by:

James M. Ortega
Professor and Chairman

Department of Applied Mathematics
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528209/AMCS85/105 Copy No. ________
July 1984

This report summarizes work under NASA Grant NAG-1-242 for the period December 1, 1983 to June 1, 1984. During this period, ten graduate students were supported. The students, their major area of interest, Langley contact, University of Virginia faculty advisor, and total period of support are summarized below.

Student	Area	Langley Contact	Advisor	Period
P. Ammann	Software Engineering	E. Senn	J. Knight	1/15/84 -
D. Bahler	Data Management	R. Fulton	J. Pfaltz	6/1/82-
S. Brilliant	Software Engineering	E. Senn	J. Knight	6/1/83 -
D. Butler	Networks	M. Holt	A. Weaver	1/1/83-5/31/84
N. Fitzgerald	Concurrent Processing	O. Storassli	T. Pratt	1/15/84 -
P. Hurst	Control Systems	R. Fulton	T. Pratt	1/1/83-5/31/84
B. LoBracco	Operating Systems	J. Shoosmith	A. Batson	7/1/83-5/31/84
E. Poole	Parallel Computing	J. Lambiotte	J. Ortega	6/1/83 -
B. Smith	Software Engineering	E.Senn	J. Knight	6/1/83 -
L. St.Jean	Software Engineering	E. Senn	J. Knight	1/1/84-1/31/84

During this period, five students who have been supported on this grant completed their master's degree. The students and the title of their thesis are summarized below.

D. Bahler - Representation of a Class of Solid Geometric Objects
D. Butler - Fault-Tolerant Protocols for Real-Time Local Area Networks
E. Overly - Generation of Control Points for an Approximating B-spline.
J. Russell - The Design of a Software Development Environment for the Intel 8748Micorcomputer
B. Thomas - Flow Through Pyramids in Dynamic Scenes

Plans for the summer of 1984 call for ten students to receive support with nine of those in residence at Langley Research Center for all or most of the summer. These students, their major area of interest, Langley contract and period at Langley are summarized below.

Student	Area	Langley Contract	Advisor	Period
P. Ammann	Software Engineering	M. Holt	J. Knight	5/28-8/3
S. Brilliant	Software Engineering	M. Holt	J. Knight	5/21-8/10
N. Fitzgerald	Concurrent Processing	O. Storassli	T. Pratt	5/21-8/17
N. Grine	Software Engineering	E. Senn	J. Knight	5/21-7/27
C. Luan	Computer Graphics	D. Lansing	W. Martin	5/21-8/17
J. Marco	Software Engineering	E. Senn	J. Knight	5/28-8/10
G. Poole	Concurrent Processing	O. Storassli	J. Ortega	5/28-8/10
B. Smith	Software Engineering	J. Senn	J. Knight	5/28-8/24
J. Taylor	Concurrent Processing	J. Lambiotte	T. Pratt	6/4-8/17

Students at Langley Research Center during Summer of 1984

We next give short summaries of the work performed during the reporting period.

Testing Method Evaluation

Paul Ammann, Masters Candidate in Computer Science
John Knight, Associate Professor of Computer Science

Several empirical tests of N version programming have been conducted or funded by NASA Langley. Currently, in progress are tests of an N version Launch Interceptor program written by graduate students at the University of Virginia and the University of California at Irvine. There are approximately thirty versions available; 120,000 pseudo-randomly generated tests have been run on nine of them.

First, it is desirable to know how complete the to date testing has been. Second, little controlled evaluation has been done on any of the formal testing methods. A mutual evaluation of a formal test method and the tested versions of the Launch Interceptor moderates both deficiencies listed. Error seeding is the chosen formal test method.

Representation of a Class of Solid Geometric Objects

Dennis R. Bahler, Ph.D. Candidate in Computer Science
John Pfaltz, Professor of Computer Science

A new representation scheme for the modeling of three-dimensional objects is defined, called the B-spline cylinder, and a general model of the geometric representation process is elaborated for use in formally evaluating this scheme. The representation is found to be unambiguous, to possess attractive design advantages, and to be quite efficient both computationally and with respect to storage. A number of shortcomings of the representation scheme have also been detected. Several unary operations are defined on the representation. An implementation of a small geometric modeling system using only objects representable by this scheme has been developed, employing hidden-surface color graphic display and a relational database management system for secondary storage.

Evaluation of Testing Methods

Susan S. Brilliant, Masters Candidate in Computer Science
John Knight, Associate Professor of Computer Science

As part of an experiment in n-version programming, a number of students at the University of Virginia and at the University of California at Irvine have independently written versions of a Pascal procedure in compliance with the same specification. As a result of extensive testing of these procedures, errors have been discovered in some of the versions. This research is directed towards determining whether testing methods that ensure that every line of a program has been executed and every branch has been executed in both directions are effective, i.e. whether the known errors in these procedures will be discovered using these testing methods.

Fault Tolerant Protocols For Real Time Local Area Networks

David W. Bulter, Masters Candidate in Computer Science
Alfred C. Weaver, Associate Professor of Computer Science

This research was an exploration of methods of enchancing fault tolerance and real–time performance of communications protocols in local area networks. The requirements for high-performance fault tolerant protocol operation were examined and a new protocol, ITP, was developed to meet these requirements. The complete data link specification of ITP was presented with analytic performance models. An ITP implementation was presented and a complete parametric study of ITP's performance in this implementation was presented. The parametric study allowed an assessment of the value of M/M/1 and M/G/1 network delay models in the performance prediction of certain real–time applications. A characterization of network faults was presented. This allowed the testing and evaluation of ITP's ability to withstand and recover from such faults to be performed. The results of these analyses showed that ITP compares favorably with currently available real–time protocols.

XFEM Implementation

Nancy J. Fitzgerald, Masters Candidate in Computer Science
Terrence Pratt, Professor of Computer Science

XFEM is a computer system intended for the solution of large scale problems in scientific and engineering computation. It is based on the use of MIMD parallel computation to achieve high computation rates. The system includes a programming environment, programming language, operating system, and machine architecture. Because the software provides an abstract "virtual machine" to the user, the precise details of the hardware and lower levels of the operating system software are not of concern to the user.

The facilities available to program the XFEM virtual computer include a base sequential language (PASCAL) with a set of additional constructs for defining tasks, for sending and receiving messages from other tasks, etc. An XFEM implementation on the Finite Element Machine (FEM) at NASA–Langley is currently underway.

Computer Performance Analysis

Bernard LoBracco, Jr., Ph.D. Candidate in Applied Mathematics
Alan Batson, Professor of Computer Science

The Computer Management Branch of the Analysis and Computation Division at NASA–Langley maintains a large number of computer systems. The emphasis of this research has been to investigate bottlenecks in the Cyber network, particularly the local disk system of the Cyber 173–A. The A machine handles only interactive jobs during normal working hours and minimization of response time is the goal.

The Cyber 173–A appears to be a network of servers to a job input to the system. The job may request CPU processing, then some disk I/O, then more cpu time, etc. A model developed by Yonathan Bard of IBM was used to determine the flow rates along various paths between the CPU and I/O devices. The key ingredient of this model is that there are multiple paths between the CPU and any of the I/O devices. Using average access rates, record lengths, and seek times, channel utilizations were estimated. By considering each access pair (CPU and an I/O device) an M/G/1 queue the mean response time can be estimated for the present and any future configurations.

Vectorizing Incomplete Cholesky Conjugate Gradient on the Cyber 203/205

Eugene L. Poole, Ph.D. Candidate in Applied Mathematics
James Ortega, Professor of Applied Mathematics

Modification of existing algorithms and development of new methods used in solving problems involving large sparse matrices on vector computers such as the Cyber 203 at NASA-Langley or experimental parallel computers such as the Finite Element Machine is one of many areas of research in large scale scientific computing today.The purpose of this research is to examine the conjugate gradient method with incomplete cholesky decomposition preconditioning to determine how effectively it can be implemented on the Cyber 203 and the Finite Element Machine.

Two programs were written to implement incomplete cholesky preconditioned conjugate gradient on the Cyber 203. Multi-colored orderings were used to achieve long vector lengths in both the decomposition and forward-back solves necessary in the incomplete cholesky preconditioning. Results have been obtained for two model problems on the Cyber 203 and compared with other preconditioning methods that have previously been used to accelerate the conjugate gradient algorithm.

Design of a Source Management System

Brandon C. Smith, Master's Candidate in Computer Science
John Knight, Associate Professor of Computer Science

The project was to design a Source Code Management System (SCMS) using the design language SDDL and to evaluate SDDL as a design tool. The need for SCMS arises because of the persistent problem in program development of controlling changes to source files. SCMS both provides a means whereby changes to the source may be maintained and controls the environment in which files may be modified.

The second portion of this project was to evaluate the design language SDDL as a design tool for the software engineer. The main purpose of SDDL is to aid the program designer by enabling him/her to produce a well formatted and structured design with such helpful features as automatic cross referencing. It was found that SDDL helped a great deal during the design process, mainly because of this automatic cross referencing and formatting. Its main drawback, and a drawback of many design tools available today, is the lack of analysis capabilities.

A LARGE SCALE EXPERIMENT IN MULTI-VERSION SOFTWARE DEVELOPMENT ABSTRACT

Lois St.Jean, Masters Candidate in Computer Science
John Knight, Associate Professor of Computer Science

A multi-version software unit incorporates two or more versions of a program which have been generated separately by programmers working from the same specifications. The multi-version unit compares results of each version and determines by a voting scheme which results to use. This method has been proposed as a means to allow software to tolerate its own faults and thereby increase its reliability.

Any reliability improvement that this method provides relies upon the fundamental assumption that versions generated independently by non–communicating programmers will fail independently. An experiment has been performed that tests the validity of this assumption. The experiment involved the preparation of detailed specifications for a medium–sized program, arranging for several versions to be written by graduate students in a software engineering class, and testing of the resulting versions. In addition, the reliability performance of a multi–version software unit was evaluated as the number of versions included in the unit increases.